

CLAIMS

- 5 1. A method of aligning bytes on a word boundary including the steps of:

storing a block of data, containing bytes which are not aligned with a word boundary, in an input register (5a);

10 determining the amount of byte misalignment with respect to the word boundary;

transferring bytes from the input register (5a) to an output register (5b), under control of a processor (1), so as to correct the byte misalignment; and

15 repeating the method on subsequent words in the data block until the data block has been transferred with alignment from an input buffer(2) to an output buffer (3);

characterised in that the processor(1):

20 (a) shifts the first n bytes of said data block into said input register (5a) in order to read the amount of byte misalignment or offset in an unused byte position of the input register (5a);

25 (b) shifts said n bytes out of said input register (5a) and adds one or more alignment offset bits and excess bits to said unused byte portion of the input register (5a), which offset bits specify the position of the word boundary, and which excess bits make up the byte in the unused byte portion; and

30 (c) reads the offset bits and excess bits in the unused byte position of the input register (5a) so as to correct for misalignment when transferring bytes from the input buffer (2) to the output buffer (3).

2. A method according to claim 1, when used to translate the encapsulation of a protocol labelled data block, wherein the data block includes a header, and the method includes the steps of:

5 removing an original header from the data block;
providing a new header;
using the steps of claim 1 to determine any byte misalignment in the
new header; and
10 transferring the new header and the original data block into storage
with any necessary shift in the data block to compensate for the byte
misalignment.

3. A method according to claim 2, in which an old trailer is also removed and
15 replaced with a new trailer, the byte alignment method of claim 1 being used to
determine any byte misalignment in the new header and in the new trailer; the new
header, the original data block and the new trailer being transferred with any
necessary shift in the data block to compensate for the byte misalignment.

20 4. Circuitry for aligning data on a word boundary, the circuitry including:

input buffer means (2) for storing a block of data including words containing bytes
which are not aligned with the word boundary;

25 output buffer means (3) for storing words containing bytes which have been aligned
with a word boundary; and a processor (1) for controlling the transfer of bytes from
an input register (5a) to an output register (5b) so as to correct for the byte
misalignment;

characterised in that the processor(1):

30 (a) shifts the first n bytes of said data block into said input register
(5a) in order to read the amount of byte misalignment or offset in an
unused byte position of the input register (5a);

(b) shifts said n bytes out of said input register (5a) and adds one or more alignment offset bits and excess bits to an unused byte portion of the input register (5a), which offset bits specify the position of the word boundary, and which excess bits make up the byte in the unused byte portion; and

(c) reads the offset bits and excess bits in the unused byte position of the input register (5a) so as to correct for misalignment when transferring bytes from the input buffer (2) to the output buffer (3). (a) input and output register means for storing n bytes of the data block;

5. Apparatus according to any of claim 4, further including means for removing an original header from a protocol labelled data block; means for providing a new header; means for determining any byte misalignment in the new header and for transferring the new header and the original data block into storage with any necessary shift in the data block to compensate for the byte misalignment.

6. Apparatus according to claim 5, in which means also included for removing an old trailer from the data block, for providing a new trailer and for determining any byte misalignment in the new header and the new trailer; the new header, new trailer and original data block being transferred into storage with any necessary shift in the data block to compensate for the byte misalignment.

7. Apparatus according to claim 5 or 6 in which the new header is constructed so that it ends on a word boundary which is complementary to that of the start of the data block and the trailer is aligned so as to match the end of the data block.

8. Apparatus according to claim 5 or 6 in which odd bytes are copied from the start and end of the data block to the new header and trailer respectively.